Analiza schematów SH-GKDS

Tomasz Rams, Piotr Pacyna (e-mail: {trams, pacyna}@kt.agh.edu.pl)
AGH Akademia Górniczo-Hutnicza w Krakowie, Katedra Telekomunikacji

STRESZCZENIE

Artykuł przybliża techniki selektywnej dystrybucji kryptograficznego klucza grupowego w kanale rozsiewczym ze stratami. Stanowi on wprowadzenie do klasy schematów Self-healing Group Key Distribution Schemes (SH-GKDS). Przedstawiono w nim najważniejsze mechanizmy wykorzystywane w projektowaniu schematów SH-GKDS, kładąc nacisk na analizę ich właściwości pod kątem bezpieczeństwa, skalowalności oraz czasu działania systemu stosującego schemat SH-GKDS.

Słowa kluczowe: *bezpieczna komunikacja grupowa, dystrybucja klucza kryptograficznego, selektywna dystrybucja klucza*

ABSTRACT

Analysis of Self-healing Group Key Distribution Schemes (SH-GKDS)

The paper presents techniques for selective group key distribution over an unreliable broadcast channel. It provides introduction to Self-healing Group Key Distribution Schemes (SH-GKDS), including guidelines for scheme design and analysis. The most important mechanisms used in SH-GKDS are described and analysed in terms of security, scalability and system life-time.

Key words: *secure group communications, group key distribution, selective key distribution*

1. Schematy SH-GKDS

W ostatnich latach obserwuje się wzrost zainteresowania komunikacją grupową ze względu na liczne zastosowania takiego modelu komunikacji. W literaturze naukowej obecne są nurty badawcze, które podejmują próbę odpowiedzi na potrzebę wprowadzenia ochrony komunikacji grupowej dla uzyskania bezpieczeństwa komunikacji pomiędzy członkami grupy. W szczególności, może być ona chroniona przy użyciu efektywnych, wydajnych obliczeniowo technik opartych na współdzielonym, symetrycznym kluczu grupowym. Niestety, raz ustanowiony klucz powinien podlegać okresowej wymianie w celu przeciwdziałania próbom kryptoanalizy. Także w przypadku zmiany składu grupy klucz powinien podlegać natychmiastowej wymianie, aby nie narazić członków grupy na utratę bezpieczeństwa. Wymianę klucza można realizować poprzez wprowadzenie systemów do selektywnej dystrybucji kryptograficznego klucza grupowego. Jednym z najbardziej obiecujących rozwiązań w tej dziedzinie jest klasa schematów Self-healing Group Key Distribution Schemes (SH-GKDS).

Schematy SH-GKDS przeprowadzają selektywną dystrybucję klucza grupowego za pośrednictwem kanału rozsiewczego, w którym mogą następować straty pakietów. Są one dostosowane do dużych i dynamicznych grup multicastowych. W schematach SH-GKDS czas jest podzielony na epoki nazywane sesjami. Na początku każdej sesji węzeł zarządzający transmituje wiadomość rozsiewczą, w celu dostarczenia klucza grupowego do wszystkich członków grupy. Każdy uprawniony użytkownik wylicza klucz grupowy na podstawie odebranej wiadomości i pewnych prywatnych informacji. Ważną właściwością schematu jest odporność na straty wiadomości. W przypadku gdy któraś z wiadomości rozsiewczych zostanie utracona, użytkownicy są w stanie wyliczyć klucz sesyjny dystrybuowany w utraconej wiadomości, na podstawie wiadomości odebranych w kolejnych sesjach, bez wysyłania żądania retransmisji do węzła zarządzającego.

Struktura artykułu jest następująca. W dalszej części rozdziału 1 opisano ogólny model schematu SH-GKDS oraz najważniejsze kryteria analizy schematów. W rozdziale 2 przedstawiono schemat bazowy wraz z analizą jego najważniejszych właściwości. Rozdział 3 zawiera opis i analizę wybranych ulepszeń, prezentowanych w odniesieniu do schematu bazowego. Rozdział 4 stanowi podsumowanie pracy. W artykule często stosowane są symbole, w celu zwiększenia przejrzystości opisów (przyjęta notacja – tab. 1).

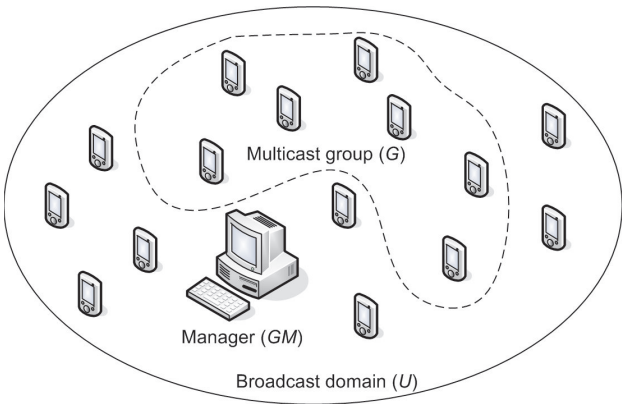
1.1. Architektura sieci

Schematy SH-GKDS stosowane są w sieciach o architekturze przedstawionej na rysunku 1. Sieć składa się z pojedynczego węzła zarządzającego (*GM*) oraz dużego zbioru węzłów użytkowników (*U*). Węzeł zarządzający dysponuje dużymi zasobami sprzętowymi, takimi jak pamięć czy moc obliczeniowa. Natomiast węzły użytkowników mają silnie ograniczone zasoby sprzętowe. Komunikacja odbywa się za pośrednictwem kanału rozsiewczego ze stratami. *GM* transmituje wiadomości broadcastowe, które są odbierane przez wszystkich użytkowników. Ze względu na mobilność węzłów oraz błędy medium transmisyjnego, niektóre wiadomości mogą zostać utracone. Retransmisja wiadomości po-

winna być stosowana jedynie w uzasadnionych przypadkach, ponieważ jest ona kosztowna i wymaga połączenia zwrotnego od odbiorców do *GM*, które może być niedostępne.

Podzbiór węzłów $G \subseteq U$ tworzy grupę multicastową. Ma ona dynamiczny skład, możliwe jest dodawanie nowych użytkowników lub usuwanie istniejących użytkowników. Głównym celem prezentowanych systemów jest zapewnienie bezpieczeństwa komunikacji pomiędzy *GM* a węzłami należącymi do grupy multicastowej *G*. Komunikacja jest chroniona przez szyfrowanie i uwierzytelnianie wiadomości, z użyciem współdzielonego symetrycznego klucza grupowego *K*. Zastosowanie współdzielonego klucza jest wygodne, ale może on zostać ujawniony przez użytkowników opuszczających grupę, lub członków grupy przechwyconych przez atakujących oraz jest narażony na kryptoanalizę. Dlatego częsta wymiana współdzielonego klucza jest konieczna do osiągnięcia wysokiego poziomu bezpieczeństwa. Klucz powinien podlegać wymianie w przypadku zmiany składu grupy *G*, a także periodycznej wymianie, w celu zmniejszenia ilości danych dostępnych do kryptoanalizy.

Częstą wymianę współdzielonego klucza można zrealizować poprzez zastosowanie bezpiecznego systemu selektywnej dystrybucji kluczy grupowych, z *GM* pełniącym rolę zaufanej strony trzeciej (Trusted Third Party).



Rys. 1. Architektura sieci

1.2. Zastosowania schematów SH-GKDS

Schematy SH-GKDS mogą być stosowane w dowolnych sieciach z kanałem rozsiewczym, w których występuje scentralizowane zarządzanie. Są one dobrze dopasowane do ochrony grup multicastowych w sieciach bezprzewodowych, np. w sieciach sensorowych, sieciach komórkowych i innych. Mogą być również używane w systemach transmisji rozsiewczej, takich jak: radio, telewizja satelitarna, telewizja kablowa, pay-per-view TV oraz serwisy informacyjne. Ze względu na dużą niezawodność oraz wysoki poziom bezpieczeństwa, schematy SH-GKDS mogą także znaleźć zastosowanie w systemach wojskowych oraz systemach bezpieczeństwa publicznego.

Przyjęta notacja wraz z wyjaśnieniami została podana w tabeli 1.

Tabela 1

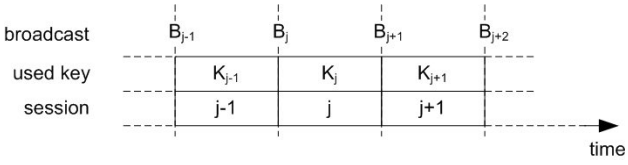
Notacja stosowana w opisie schematów SH-GKDS

GM	węzeł zarządzający
U	zbiór wszystkich użytkowników należących do sieci
U_i	*i*-ty użytkownik
K_j	klucz sesyjny dystrybuowany w sesji *j*
B_j	wiadomość rozsiewcza transmitowana w sesji *j*
S_i	klucz personalny użytkownika U_i
G_j	zbiór pełnoprawnych członków grupy multicastowej w sesji *j*
R_j	zbiór użytkowników wykluczonych z grupy w sesji *j*
$R_{<1,j>}$	zbiór wszystkich użytkowników wykluczonych w sesji *j* lub wcześniej, tj. $R_{<1,j>} = R_j \cup R_{j-1} \cup ... \cup R_2$
I_U	zbiór indeksów przypisanych wszystkim użytkownikom należącym do sieci, gdzie x_i jest przypisany użytkownikowi U_i, tj. $I_U = \{x_i \in F_q\}_{U_i \in U}$
I_{G_j}	zbiór indeksów przypisanych wszystkim członkom grupy G_j, tj. $I_{G_j} = \{x_i \in F_q\}_{U_i \in G_j}$
I_{R_j}	zbiór indeksów przypisanych użytkownikom wykluczonym w sesji *j*, tj. $I_{R_j} = \{x_i \in F_q\}_{U_i \in R_j}$
$I_{R_{<1,j>}}$	zbiór indeksów przypisanych użytkownikom wykluczonym w sesji *j* lub wcześniej, tj. $I_{R_{<1,j>}} = I_{R_j} \cup \in I_{R_{j-1}} \cup \ldots \cup I_{R_2}$
t	maksymalna liczba użytkowników, których można wykluczyć z grupy
m	maksymalna liczba sesji

1.3. Ogólny model systemu SH-GKDS

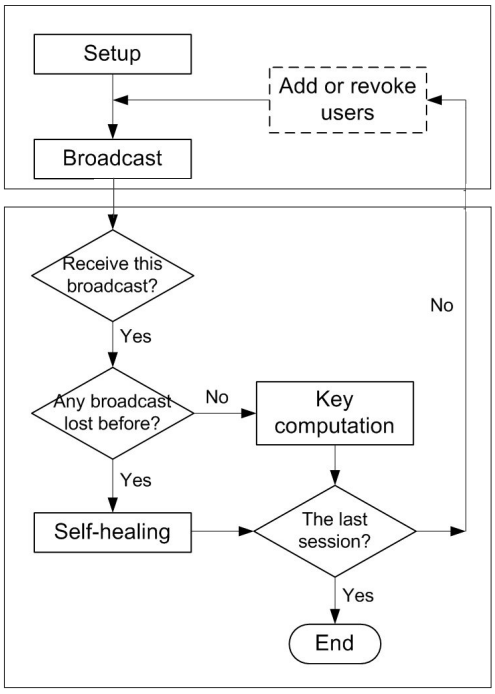
W schematach SH-GKDS czas istnienia grupy multicastowej jest podzielony na epoki zwane sesjami. W każdej sesji używany jest inny klucz grupowy, nazywany kluczem sesyjnym (rys. 2). Skład grupy multicastowej może ulegać zmianie w kolejnych sesjach. Na początku sesji *j*, węzeł zarządzający przeprowadza dystrybucję nowego klucza grupowego K_j do wszystkich użytkowników należących do grupy G_j. Czas trwania pojedynczej sesji jest wyznaczany przez *GM* w zależności od zmian w składzie grupy multicastowej oraz zachowania węzłów, a także na podstawie przyjętej polityki bezpieczeństwa. Aby bezpiecznie usunąć użytkownika U_i z grupy multicastowej G_j, trzeba rozpocząć nową sesję $j + 1$, w której użytkownicy należący do grupy $G_{j+1} = G_j \backslash \{U_i\}$ będą stosować nowy klucz K_{j+1}do

ochrony komunikacji grupowej, który jest niedostępny dla U_i. Bezpieczne dodanie nowego użytkownika również wymaga rozpoczęcia nowej sesji, aby uniemożliwić mu dostęp do wiadomości wymienianych przed przyjęciem go do grupy. Ponadto klucz sesyjny powinien być odświeżany z pewną minimalną częstotliwością, gwarantującą, że atakujący nie będzie w stanie zebrać wystarczającej ilości danych potrzebnych do przeprowadzenia skutecznej kryptoanalizy. Dlatego wybór długości sesji stanowi kompromis pomiędzy obciążeniem wynikającym z częstej dystrybucji kluczy oraz wymaganym poziomem bezpieczeństwa.



Rys. 2. Podział czasu na sesje

Na początku każdej sesji *j* węzeł narządzający *GM* transmituje wiadomość rozsiewczą B_j, aby dostarczyć nowy klucz sesyjny K_j do wszystkich użytkowników należących do grupy G_j. Ponieważ B_j jest przesyłana za pośrednictwem kanału rozsiewczego, może być ona odebrana przez wszystkie węzły należące do U, również węzły nieuprawione do poznania K_j. Dlatego K_j musi być ukryte w B_j za pomocą dodatkowych danych maskujących, aby zapobiec nieuprawnionemu dostępowi do K_j.



Rys. 3. Schemat blokowy systemu SH-GKDS (źródło [27])

Schematy dystrybucji kluczy grupowych należące do klasy *Self-healing Group Key Distribution Schemes* (SH-GKDS) mają następujące właściwości:

- *selektywna dystrybucja klucza grupowego* – dystrybucja klucza K_j jest przeprowadzana w taki sposób, że tylko pełnoprawni członkowie grupy G_j są w stanie wyliczyć K_j,
- *dynamiczne wykluczanie użytkowników* – istnieje możliwość usunięcia z grupy dowolnego użytkownika,
- *dynamiczne przyłączanie nowych użytkowników* – istnieje możliwość dodawania nowych użytkowników w trakcie działania systemu,
- *zdolność self-healing* – schemat jest odporny na błędy komunikacyjne dzięki wprowadzeniu redundancji w wiadomościach B_j.

Schemat SH-GKDS składa się zwykle z czterech podstawowych procedur, wykonywanych w kolejności przedstawionej na rysunku 3. Są to:

- *Setup*: *GM* generuje klucz personalny S_i dla każdego użytkownika U_i należącego do grupy multicastowej i dostarcza go za pomocą bezpiecznego kanału komunikacyjnego. Klucz personalny S_i stanowi sekretną informację, znaną jedynie węzłowi U_i i używaną przez niego do odzyskiwania przyszłych kluczy sesyjnych z odebranych wiadomości rozsiewczych. Węzły mogą być wyposażone w klucze personalne przed wdrożeniem systemu, albo w trakcie przyłączania się do grupy multicastowej.
- *Broadcast*: W sesji *j* węzeł zarządzający tworzy i transmituje wiadomość B_j o następujących właściwościach:
  - każdy członek grupy $U_i \in G_j$ jest w stanie wyliczyć z niej klucz grupowy K_j,
  - pozostałe węzły nie są w stanie uzyskać z niej żadnych użytecznych informacji na temat K_j.
- *Session key calculation*: Każdy $U_i \in G_j$ używa swojego klucza personalnego S_i do wyliczenia K_j z odebranej wiadomości rozsiewczej: $K_j = \eta(B_j, S_i)$.
- *Self-healing*: Użytkownik U_i, będący pełnoprawnym członkiem grupy, który poprawnie odebrał dwie wiadomości B_l oraz B_r, takie że $l < r$, jest w stanie odzyskać wszystkie klucze sesyjne $K_l, ..., K_r$, wyliczając: $K_j = \zeta(B_l, B_r, S_i)$, gdzie $l \le j \le r$.

1.4. Kryteria analizy schematów

Istniejące rozwiązania stanowią zwykle kompromis pomiędzy: skalowalnością, bezpieczeństwem oraz czasem działania systemu. Skalowalność systemu jest określona przez obciążenie komunikacyjne powodowane przez działający schemat dystrybucji kluczy oraz przez wymagania pamięciowe stawiane węzłom użyt-

kowników. Jako metrykę obciążenia komunikacyjnego przyjmuje się rozmiar wiadomości rozsiewczej, natomiast metryką wymagań pamięciowych jest rozmiar klucza personalnego przechowywanego przez użytkownika. Czas życia systemu jest wyznaczony przez maksymalną liczbę sesji oraz maksymalną liczbę użytkowników których można wykluczyć z grupy.
Analiza bezpieczeństwa schematów SH-GKDS sprowadza się zwykle do weryfikacji poufności dystrybuowanego klucza. Przyjmijmy, że m oznacza maksymalną liczbę sesji, a t jest parametrem określającym gwarantowany poziom bezpieczeństwa systemu. Schemat jest traktowany jako bezpieczny, jeśli wykazuje następujące właściwości:

- *t-forward secrecy* – dowolny zbiór $R_{<1,j>} \subseteq U$ składający się z użytkowników wykluczonych z grupy przed oraz w trakcie sesji j, taki że $|R_{<1,j>}| \leq t$, nie jest w stanie współpracując ze sobą wyliczyć żadnego z przyszłych kluczy grupowych K_j, ..., K_m.
- *t-backward secrecy* – dowolny zbiór $J_{<j+1,m>} \subseteq U$ składający się z użytkowników dodanych do grupy po sesji j, taki że $|J_{<j+1,m>}| \leq t$, nie jest w stanie współpracując ze sobą wyliczyć żadnego z przeszłych kluczy grupowych K_1, ..., K_j.
- *t-collusion resistance* – przy zakłożeniu że $R_{<1,l>} \subseteq U$ jest zbiorem użytkowników wykluczonych z grupy przed sesją oraz w trakcie sesji l, natomiast $J_{<r+1,m>} \subseteq U$ jest zbiorem użytkowników dodanych do grupy po sesji r, oraz że $|R_{<1,l>} \cup J_{<r+1,m}| \leq t$, to współpracująca koalicja tych zbiorów nie jest w stanie wyliczyć żadnego z kluczy grupowych K_l, ..., K_r.

2. Schemat bazowy

Pierwszy schemat SH-GKDS został zaproponowany przez Staddon i in. w pracy [1]. Następnie został on ulepszony i znacznie uproszczony przez Liu i in. w publikacji [2] oraz Blundo i in. w artykule [3]. Schematy te stanowią podstawę licznych dalszych prac, mających na celu poprawę właściwości systemów dystrybucji kluczy grupowych. W naszym artykule posłużymy się schematem [3] jako punktem wyjścia do opisu i analizy najważniejszych rozwiązań zaproponowanych w tej dziedzinie.
Schemat zaproponowany przez Blundo i in. opiera się na arytmetyce wielomianowej. Wykorzystuje on wielomiany stopnia t zdefiniowane na ciele skończonym F_q, aby osiągnąć selektywną dystrybucję kluczy sesyjnych. Zastosowanie wielomianów pozwala na zmniejszenie ilości danych, które muszą być wytransmitowane w celu bezpiecznego dostarczenia materiału do wyliczenia klucza sesyjnego do uprawnionych węzłów. Klucze personalne użytkowników nie są wybierane w sposób losowy, lecz są ze sobą powiązane, co pozwala na osiągnięcie kompromisu pomiędzy bezpieczeństwem a wprowadzanym obciążeniem komunikacyjnym. Wszystkie opisywane obliczenia są prowadzone na ciele skończonym F_q, gdzie q jest dużą liczbą pierwszą.
Schemat 3 zaprezentowany w [3], nazywany w dalszej części artykułu *schematem bazowym*, może być opisany w następujący sposób:

- *Setup*: *GM* generuje w losowy i niezależny sposób m wielomianów maskujących $s_1(x)$, ..., $s_m(x) \in F_q[x]$, z których każdy jest stopnia t, oraz m kluczy sesyjnych K_1, ..., $K_m \in F_q$. Następnie przypisuje każdemu użytkownikowi $U_i \in U$ losowy indeks $x_i \in F_q$. W trakcie inicjalizacji systemu każdy członek grupy $U_i \in G_1$ otrzymuje od *GM* swój klucz personalny $S_i = [x_i, s_1(x_i), ..., s_m(x_i)]$ za pośrednictwem bezpiecznego kanału komunikacyjnego.
- *Broadcast*: W sesji j węzeł zarządzający wybiera zbiór losowych unikalnych indeksów $W_j = \{w_1,..., w_t\} \subseteq F_p$, taki że $|W_j| = t$, $I_{R_{<1,j>}} \subseteq W_j$, $I_{G_j} \cap W_j = \varnothing$ oraz $0 \notin W_j$. Następnie *GM* wylicza pojedynczy fragment danych

  $$b_j = \left[z_j = K_j + s_j(0), \{(w_l, s_j(w_l))\}_{w_l \in W_j} \right]$$

  oraz transmituje wiadomość rozsiewczą

  $$B_j = \left[b_j, b_{j-1}, ..., b_1 \right]$$

- *Session key calculation*: Każdy użytkownik $U_i \in G_j$, który poprawnie odebrał wiadomość B_j, jest w stanie wyliczyć klucz sesyjny K_j według następującej procedury:
  - Odzyskuje wartość $s_j(0)$ przez interpolację Lagrange'a na podstawie t punktów $\{(w_l, s_j(w_l))\}_{w_l \in W_j}$ otrzymanych w b_j, oraz pojedynczego punktu $(x_i, s_j(x_i))$ zawartego w jego kluczu personalnym.
  - Wylicza klucz sesyjny $K_j = z_j - s_j(0)$.
- *Self-healing*: Użytkownik U_i, który poprawnie odebrał wiadomość B_j, jest w stanie odzyskać brakujący klucz sesyjny K_l, dystrybuowany w dowolnej poprzedniej sesji $l \leq j$, jeśli tylko był on pełnoprawnym członkiem grupy G_l, ponieważ B_j zawiera b_l.

Schemat zakłada, że można przewidzieć maksymalną liczbę sesji, jaka będzie potrzebna w trakcie działania systemu oraz maksymalną liczbę użytkowników, których będzie można wykluczyć z grupy. Maksymalna liczba sesji, oznaczana parametrem m, wpływa na liczbę wielomianów maskujących $s_j(x)$, a tym samym liczbę punktów $s_j(x_i)$ zawartych w kluczu personalnym użytkownika. Z kolei maksymalna liczba użytkowników, których można wykluczyć, oznaczana parametrem t,

jest ograniczona przez stopień zastosowanych wielomianów. Oba parametry muszą być wyznaczone już w trakcie inicjalizacji systemu.

Klucz sesyjny K_j jest ukryty przez punkt na wielomianie maskującym $s_j(0)$. Użytkownik U_i musi interpolować wielomian $s_j(x)$ aby wyliczyć $s_j(0)$ i odtworzyć K_j. Żeby móc poprawnie przeprowadzić interpolację wielomianu stopnia t, użytkownik U_i musi znać co najmniej t+1 różnych punktów na tym wielomianie. Wszyscy użytkownicy należący do $R_{<1,j>}$ mają dostęp jedynie do t różnych punktów, ponieważ $I_{R_j} \subseteq W_j$, a w związku z tym punkty zawarte w ich kluczach personalnych znajdują się już w zbiorze $\{(w_l, s_j(w_l))\}_{w_l \in W_j}$ przesyłanym w komunikacie rozsiewczym b_j. Nie są oni w stanie uzyskać żadnych informacji na temat K_j, nawet jeśli ze sobą współpracują, a więc schemat zapewnia *t-forward secrecy*.

Użytkownicy dodani do grupy multicastowej w sesji j są wyposażani w klucz personalny $S_i = [x_i, s_j(x_i), ..., s_m(x_i)]$. Nie otrzymują oni punktów na wielomianach maskujących stosowanych w poprzednich sesjach, a w związku z tym nie są w stanie uzyskać żadnych informacji na temat przeszłych kluczy sesyjnych. Dlatego schemat zapewnia *backward secrecy* oraz *t-collusion resistance*.

Odporność schematu na utratę wiadomości została osiągnięta przez każdorazową retransmisję wszystkich poprzednich fragmentów danych $b_1, ..., b_{j-1}$ w nowej wiadomości rozsiewczej $B_j = [B_{j-1}, b_j]$. Rozmiar wiadomości rozsiewczej transmitowanej w sesji j wynosi $(2tj + j)\log q$ bitów, natomiast rozmiar klucza personalnego przydzielanego użytkownikowi, który został dodany do grupy w sesji j, wynosi $(m - j + 1)\log q$ bitów.

3. Opis i analiza wybranych ulepszeń

W ostatnich latach pojawiło się wiele propozycji ulepszeń schematów dystrybucji kluczy grupowych. W dalszej części artykułu prezentujemy najważniejsze rozwiązania oraz przeprowadzamy analizę ich właściwości w odniesieniu do schematu bazowego, zaprezentowanego w rozdziale 2.

3.1. Wielokrotne użycie wielomianu maskującego

W pracach [2] i [4] zaproponowano wykorzystanie tak zwanego *wielomianu wykluczającego* w celu realizacji selektywnej dystrybucji klucza grupowego. Ze względu na swoją prostotę, mechanizm ten zyskał dużą popularność w schematach SH-GKDS. Użytkownik $U_i \in G_1$ otrzymuje klucz personalny $S_i = [x_i, h_1(x_i), ..., h_m(x_i)]$, gdzie $h_j(x)$ jest wielomianem maskującym stopnia t, wygenerowanym przez *GM* w czasie inicjalizacji systemu. Pojedynczy fragment danych transmitowany w wiadomościach rozsiewczych to:

$$b_j = \left[I_{R_{<1,j>}}, P_j(x) = r_j(x) \cdot K_j + h_j(x) \right]$$

gdzie $r_j(x)$ jest nazywany wielomianem wykluczającym i jest zdefiniowany jako $r_j(x) = \Pi_{x_i \in I_{R_{<1,j>}}}(x - x_i)$. Użytkownik $U_i \in G_j$, który poprawnie odebrał wiadomość B_j, a tym samym fragment danych b_j, może wyliczyć klucz sesyjny w następujący sposób:

$$K_j = \frac{P_j(x_i) - h_j(x_i)}{r_j(x_i)}$$

Selektywny dostęp do klucza sesyjnego K_j jest osiągany przez połączenie wielomianu wykluczającego $r_j(x)$ i wielomianu maskującego $h_j(x)$. Znajomość tylko jednego punktu na wielomianie maskującym sprawia, że użytkownik U_i może jedynie wyliczyć wartość $r_j(x) \cdot K_j$ dla argumentu $x = x_i$. Dla wszystkich wykluczonych użytkowników $U_i \in R_{<1,j>}$, wielomian $r_j(x)$ przyjmuje wartość $r_j(x_i) = 0$, dlatego nie są oni w stanie uzyskać żadnych informacji o K_j.

Schematy [2] i [4] oparte na wielomianie wykluczającym mają właściwości bardzo podobne do schematu bazowego. Osiągają taki sam poziom bezpieczeństwa oraz mają bardzo zbliżony koszt komunikacyjny i wymagania pamięciowe. Posiadają również takie same ograniczenia na maksymalną liczbę sesji oraz maksymalną liczbę użytkowników których można wykluczyć. Dlatego nie stanowią one wyraźnego ulepszenia w stosunku do schematu bazowego.

Dutta i in. w pracy [5] zasugerowali, że można używać tego samego wielomianu maskującego $h(x)$ w wielu sesjach, a dzięki temu można stworzyć schemat o nieograniczonym czasie działania. Pomysł ten opiera się na założeniu, że dystrybucja pojedynczego klucza sesyjnego K_j nie powoduje ujawnienia żadnych informacji na temat wielomianu maskującego. Założenie to jest niestety nieprawdziwe, a w związku z tym zaproponowany schemat nie spełnia podstawowych wymagań bezpieczeństwa. Łatwo udowodnić, że w trakcie dystrybucji K_j wielomian $h(x)$ jest ujawniany wszystkim użytkownikom $U_i \in G_j$. Znając K_j, użytkownik jest w stanie wyliczyć $h(x) = P_j(x) - r_j(x) \cdot K_j$. Analogiczna sytuacja ma również miejsce w przypadku schematu bazowego. Dlatego w żadnym z tych schematów nie można wielokrotnie używać pojedynczego wielomianu maskującego.

Schemat zaproponowany w [6] znacznie obniżył rozmiar kluczy personalnych przechowywanych przez użytkowników, przez zastosowanie wielomianu dwóch zmiennych $\psi(x, y) \in F_q[x, y]$, stopnia t dla każdej zmiennej. Poprawa właściwości schematu wynika z założenia, że wielomian $\psi(x, y)$ może być wykorzystany do generowania ciągu losowych wielomianów jednej

zmiennej, używanych jako wielomiany maskujące. Użytkownik $U_i \in G_1$ otrzymuje klucz personalny $S_i = [x_i, \psi(x_i, y)]$. W sesji j węzeł zarządzający generuje nowy wielomian maskujący $h_j(x) = \psi(x, \alpha_j)$, gdzie α_j jest pewną predefiniowaną stałą, a użytkownik U_i wylicza swój punkt na wielomianie maskującym jako $h_j(x_i) = \psi(x_i, \alpha_j)$. Niestety przyjęte założenie o losowości generowanych wielomianów jest nieprawdziwe i kolejne wielomiany są wyraźnie od siebie zależne. Jak już wspomniano, dystrybucja klucza K_j powoduje ujawnienie $h_j(x)$ wszystkim użytkownikom $U_i \in G_j$. W związku z tym, użytkownik który jest pełnoprawnym członkiem grupy multicastowej przez co najmniej t+1 sesji, jest w stanie odtworzyć wielomian $\psi(x, y)$ na drodze interpolacji Lagrange'a. Szczegóły tego ataku zostały po raz pierwszy opisane w artykule [7]. Ten sam mechanizm został zastosowany w kilku późniejszych pracach, między innymi [8–11]. Należy podkreślić, że wszystkie te schematy są bezpieczne jedynie w ciągu pierwszych t sesji.

3.2. Wielomian dostępowy

Mechanizm selektywnej dystrybucji klucza grupowego oparty na tak zwanym *wielomianie dostępowym* został wprowadzony przez Zou i in. w pracy [12], a następnie został uproszczony i ulepszony w artykule [13]. Zyskał on w ostatnich latach duże zainteresowanie badaczy, jako metoda pozwalająca na wykluczanie dowolnej liczby użytkowników i umożliwiająca stworzenie schematów o nieograniczonej liczbie sesji. Mechanizm ten wykorzystuje wielomian $a(x)$, który jest skonstruowany w taki sposób, że dla uprawnionych członków grupy multicastowej przyjmuje wartość 1, natomiast dla wszystkich innych przyjmuje pewną losową wartość. Użytkownik $U_i \in G_1$ otrzymuje klucz personalny $S_i = [x_i, h(x_i)]$, gdzie $x_i \in F_q$ jest unikalnym, sekretnym indeksem przypisanym użytkownikowi U_i, a $h(x)$ jest wielomianem maskującym stopnia t, wygenerowanym przez *GM* w czasie inicjalizacji systemu. Pojedynczy fragment transmitowanych danych to:

$$b_j = \left[P_j(x) = a_j(x) \cdot K_j + h(x) \right]$$

gdzie $a_j(x)$ jest nazywany wielomianem dostępowym i jest zdefiniowany jako $a_j(x) = (x - r_j) \prod_{x_i \in I_{G_j}} (x - x_i) + 1$, a $r_j \in F_q \setminus I_U$ jest losowo wybraną wartością. Użytkownik $U_i \in G_j$, który poprawnie odebrał wiadomość B_j, a tym samym fragment danych b_j, może wyliczyć klucz sesyjny w następujący sposób:

$$K_j = P_j(x_i) - h(x_i)$$

Znajomość tylko jednego punktu na wielomianie maskującym sprawia, że użytkownik U_i może jedynie wyliczyć wartość $a_j(x) \cdot K_j$ dla argumentu $x = x_i$. Dla wszystkich nieuprawnionych użytkowników $U_i \in U \setminus G_j$, wielomian $a_j(x)$ przyjmuje losowe wartości, dlatego nie są oni w stanie uzyskać żadnych informacji o K_j.

Należy zauważyć, że stopień wielomianu $a_j(x)$ jest równy $|G_j| + 1$, więc dla grup o rozmiarze większym lub równym t wielomian maskujący $h(x)$ nie ukrywa go w całości. Załóżmy, że $(\alpha_{|Gj|+1, \ldots,} \alpha_0)$ są współczynnikami wielomianu $a_j(x) = \alpha_{|G_j|+1} x^{|G_j|+1} + \alpha_{|G_j|} x^{|G_j|} + \ldots + \alpha_1 x + \alpha_0$, natomiast $(h_t, \ldots, h_0)$ to współczynniki wielomianu $h(x) = h_t x^t + h_{t-1} x^{t-1} + \ldots + h_1 x + h_0$. Współczynniki wielomianu $P_j(x)$, który jest transmitowany w b_j, mają następujące wartości $(p_{|Gj|+1} = \alpha_{|Gj|+1} K_j, \ldots, p_{t+1} = \alpha_{t+1} K_j, p_t = \alpha_t K_j + h_t, \ldots, p_0 = \alpha_0 K_j + h_0)$. Ponieważ współczynnik $\alpha_{|G_j|+1}$ jest zawsze równy 1, każdy kto odbierze wiadomość B_j, a tym samym fragment danych b_j, jest w stanie wyliczyć klucz sesyjny $K_j = p_{|G_j|+1}$. W związku z tym schematy oparte na wielomianie dostępowym, zrealizowanym w opisany wcześniej sposób, nie spełniają podstawowych wymagań bezpieczeństwa w przypadku gdy grupa multicastowa jest większa lub równa t.

Autorzy wielu rozwiązań opartych na wielomianie dostępowym, między innymi [12–15], sugerują że dzięki wprowadzeniu poufności indeksów przypisanych poszczególnym użytkownikom, można stworzyć schematy o nieograniczonej liczbie sesji. Schematy te są oparte na założeniu, że dystrybucja pojedynczego klucza sesyjnego K_j nie powoduje ujawnienia żadnych informacji na temat wielomianu $h(x)$ albo zbioru I_{G_j} i w związku z tym można bezpiecznie używać tego samego wielomianu $h(x)$ w wielu sesjach. Niestety założenie to jest nieprawdziwe, ponieważ dystrybucja klucza sesyjnego powoduje częściowy wyciek informacji o $h(x)$ oraz I_{G_j}, które mogą być później wykorzystane do uzyskania nieuprawionego dostępu do dystrybuowanych kluczy, co zostanie przez nas udowodnione w osobnej publikacji. Dlatego mechanizmy selektywnej dystrybucji klucza grupowego oparte na wielomianie dostępowym nie mogą być użyte do zrealizowania schematu o nieograniczonej liczbie sesji.

Yuan i in. zaproponowali w [16] schemat oparty na wielomianie dostępowym, który jest bezpieczny i pozwala na wykluczenie z grupy dowolnej liczby użytkowników. W celu zapewnienia bezpieczeństwa, w każdej sesji użyto innego wielomianu maskującego $h_j(x)$, a maksymalny rozmiar grupy multicastowej został ograniczony do t. Schemat ten osiąga taki sam poziom bezpieczeństwa jak schemat bazowy. Niestety może on być stosowany jedynie w specyficznych środowiskach, w których liczba węzłów wykluczonych jest znacznie większa od liczby członków grupy multicastowej. Dla dużych grup schemat ten wprowadza znaczne obciążenie komunikacyjne, ponieważ rozmiar B_j jest proporcjonalny do rozmiaru grupy G_j, a w związku z tym schemat nie jest skalowalny.

3.3. Vector Space Secret Sharing

Schemat bazowy opisany w rozdziale 2, oparty jest na algorytmie współdzielenia sekretu Shamir's Secret Sharing (SSS). W publikacjach [17–20] zaproponowano mechanizmy selektywnej dystrybucji kluczy grupowych oparte na algorytmie *Vector Space Secret Sharing* (VSS). Algorytm VSS stanowi uogólnienie algorytmu SSS, które działa na ogólnej monotonicznie rosnącej strukturze dostępowej i dzięki temu pozwala osiągnąć większą elastyczność schematu dystrybucji kluczy. W rzeczywistości jednak parametry zastosowanej struktury dostępowej muszą być określone już w trakcie inicjalizacji systemu, ze względu na konieczność wyliczenia kluczy personalnych użytkowników.
Według naszej wiedzy, nie zaproponowano jeszcze żadnej struktury dostępowej, która pozwoliłaby na osiągnięcie wyraźnie lepszych właściwości schematu niż w przypadku zastosowania algorytmu SSS. W efekcie schematy oparte na VSS, mają właściwości bardzo zbliżone, a często identyczne jak schemat bazowy. Podobnie jak w przypadku schematu bazowego, wymagają one wyznaczenia w czasie inicjalizacji maksymalnej liczby sesji oraz maksymalnej liczby użytkowników, których można wykluczyć z grupy.

3.4. Arytmetyka wykładnicza

Schematy oparte na arytmetyce wykładniczej zostały zaproponowane w artykule [1], a następnie ulepszone w pracy [3]. Mogą być one traktowane jako rozszerzenie schematów opartych na wielomianach, ponieważ przenoszą obliczenia prowadzone na wielomianach do wykładników. Dzięki temu, w trakcie dystrybucji klucza sesyjnego nie są ujawniane żadne informacje na temat kluczy personalnych oraz stosowanych wielomianów maskujących. Bezpieczeństwo schematu opiera się na trudności rozwiązania problemu *Decisional Diffie Hellman* (DDH). Przyjmijmy że F_p jest ciałem skończonym stopnia p, gdzie p jest dużą liczbą pierwszą, natomiast g jest generatorem cyklicznej podgrupy $H \subseteq F_q^*$ stopnia p w której problem *DDH* jest uznawany za nierozwiązywalny. Użytkownik $U_i \in G_1$ otrzymuje klucz personalny $S_i = [x_i, s(x_i)]$, gdzie $s(x)$ jest wielomianem stopnia t wygenerowanym przez *GM* w czasie inicjalizacji systemu. Pojedynczy fragment transmitowanych danych to:

$$b_j = \left[g^{v_j},\ z_j = g^{\kappa_j + v_j \cdot s(0)},\ \left\{ \left(w_l,\ g^{v_j \cdot s(w_l)} \right) \right\}_{w_l \in W_j} \right]$$

gdzie $W_j = \{w_1, ..., w_t\} \subseteq F_p$ jest zbiorem losowych unikalnych indeksów, takich że $|W_j| = t$, $I_{R_{<1,j>}} \subseteq W_j$, $I_{G_j} \cap W_j = \varnothing$ oraz $0 \notin W_j$, natomiast $v_j \in F_p$ jest sekretną wartością losowo wygenerowaną przez *GM*. Użytkownik $U_i \in G_j$, który poprawnie odebrał wiadomość B_j, a tym samym fragment danych b_j, może wyliczyć wartość $g^{v_j \cdot s(0)}$ przeprowadzając interpolację w wykładnikach na podstawie t punktów $\{(w_l, g^{v_j \cdot s(w_l)})\}_{w_l \in W_j}$ oraz pojedynczego punktu $(x_i, g^{v_j \cdot s(x_i)})$ uzyskanego z jego klucza personalnego. Następnie wylicza klucz sesyjny $K_j = g^{\kappa_j} = z_j / g^{v_j \cdot s(0)}$.
Jak łatwo zauważyć, schemat ten jest realizacją schematu bazowego w arytmetyce wykładniczej. W odróżnieniu od schematu bazowego, pozwala on na przeprowadzenie dowolnej liczby sesji, ponieważ umożliwia bezpieczne użycie tego samego wielomianu maskującego $s(x)$ w wielu sesjach. W trakcie dystrybucji klucza K_j użytkownicy $U_i \in G_j$ poznają jedynie wartości $g^{v_j \cdot s(x)}$, które ze względu na problem DDH nie dają żadnych informacji o $s(x)$. Schemat pozwala na wykluczenie maksymalnie t użytkowników w ciągu całego czasu działania systemu. Można wykazać, że osiąga on *t-forward secrecy*, przeprowadzając rozumowanie analogiczne do analizy schematu bazowego.
Niestety, ze względu na użycie we wszystkich sesjach tego samego wielomianu maskującego $s(x)$, klucze personalne użytkowników nie zależą od numeru sesji w której użytkownik został dodany do grupy. U_i dodany w sesji j otrzymuje klucz personalny $S_i = [x_i, s(x_i)]$ i w związku z tym jest w stanie wyliczyć wszystkie klucze sesyjne dystrybuowane w poprzednich sesjach. Z tego względu schematy oparte na arytmetyce wielomianowej nie zapewniają właściwości *backward secrecy,* a co za tym idzie, nie zapewniają również *collusion resistance*. Nie spełniają więc wszystkich wymagań bezpieczeństwa.

3.5. Bilinear pairing

W pracach [21–23] zaproponowano schematy SH-GKDS oparte na *Pairing-Based Cryptography*. W schematach tych każdemu użytkownikowi $U_i \in U$ jest przydzielona para kluczy: publiczny i prywatny. U_i używa klucza prywatnego, aby wyliczyć klucz sesyjny z odebranej wiadomości rozsiewczej B_j. W dużym uproszczeniu można przyjąć, że transmitowany fragment danych b_j zawiera kryptogramy, uzyskane przez zaszyfrowanie klucza sesyjnego K_j kluczem publicznym każdego użytkownika $U_i \in G_j$. Rzeczywiste schematy są znacznie bardziej skomplikowane, my jednak nie będziemy szczegółowo ich opisywać, lecz skoncentrujemy się na zaprezentowaniu ich najważniejszych właściwości.
Schematy te posiadają unikalne właściwości, rozwiązujące większość problemów występujących w schematach opartych na wielomianach. Ten sam klucz prywatny użytkownika może być używany we wszystkich sesjach, ponieważ nie jest on ujawniany w czasie dystrybucji klucza sesyjnego. Ponadto, istnieje możliwość wykluczenia dowolnej liczby użytkowników, a schematy zachowują przy tym *backward secrecy*, *forward secrecy* i *collusion resistance*. Można więc po-

wiedzieć, że są to schematy o nieograniczonym czasie działania.

Niestety w tej klasie rozwiązań występuje bardzo duże obciążenie komunikacyjne, wynikające z dystrybucji kluczy grupowych. Rozmiar wiadomości rozsiewczej zależy od całkowitej liczby członków grupy multicastowej i jest zwykle rzędu $O(|G_j|)$. W związku z tym schematy te nie są skalowalne i mogą być praktycznie stosowane jedynie w grupach o bardzo małych rozmiarach. Również koszt obliczeniowy, związany z wyliczeniem przez użytkownika klucza sesyjnego z wiadomości rozsiewczej, jest znacznie wyższy niż w przypadku schematów opartych na wielomianach.

3.6. Łańcuchy mieszające

Wykorzystanie łańcuchów mieszających do realizacji mechanizmu *self-healing* zostało po raz pierwszy zaproponowane w artykułach [24] i [25]. Główna idea polega na wprowadzeniu zależności pomiędzy kolejnymi kluczami sesyjnymi, które pozwoliłyby na odzyskanie utraconych kluczy sesyjnych bez konieczności retransmisji wszystkich przeszłych b_1, ..., b_{j-1} w każdej wiadomości rozsiewczej B_j. Łańcuchem mieszającym nazywana jest sekwencja wartości $(x_0, x_1, ..., x_n)$, gdzie x_0 jest elementem startowym łańcucha, a kolejne elementy są wyliczane przez wielokrotne zastosowanie bezpiecznej jednokierunkowej funkcji skrótu H: $F_q \rightarrow F_q$, tj. ($x_1 = H(x_0)$, $x_2 = H(x_1)$, ..., $x_n = H(x_{n-1})$). Znając dowolny element x_j: $0 < j < n$ nie można wyliczyć żadnego elementu x_l, takiego że $0 \leq l < j$, natomiast łatwo jest wyliczyć dowolny element x_r, taki że $j < r \leq n$, ponieważ $x_r = H^{r-j}(x_j)$, gdzie $H^j(\cdot)$ oznacza że funkcja H została zastosowana i razy. Mechanizm zaproponowany w [25] opiera się na dwóch łańcuchach mieszających: *forward chain* $(K_0^F, K_1^F, ..., K_m^F)$ oraz *backward chain* $(K_0^B, K_1^B, ..., K_m^B)$. Łańcuchy są wyliczane przez *GM* w czasie inicjalizacji systemu, przez wielokrotne zastosowanie funkcji skrótu na losowo wybranych wartościach startowych $K_0^F \in F_q$ oraz $K_0^B \in F_q$, tj. odpowiednio $(K_1^F = H(K_0^F), ..., K_m^F = H^m(K_0^F))$ oraz $(K_1^B = H(K_0^B), ..., K_m^B = H^m(K_0^B))$. Użytkownik $U_i \in G_1$ otrzymuje klucz personalny $S_i = [K_1^F, x_i, s_1(x_i), ..., s_m(x_i)]$. Klucz sesyjny jest zdefiniowany jako $K_j = K_j^F + K_{m-j+1}^B$. W sesji j, węzeł zarządzający transmituje element K_{m-j+1}^B. z wykorzystaniem jednego z opisanych wcześniej algorytmów selektywnej dystrybucji, w taki sposób, że jedynie użytkownicy $U_i \in G_j$ są w stanie go odebrać. Węzeł $U_i \in G_j \cap G_r$, który poprawnie odebrał wiadomość B_r, taką że $j < r$, i wyliczył K_{m-r+1}^B, jest w stanie odzyskać zagubiony klucz K_j wyliczając $K_j = H^{r-j}(K_{m-r+1}^B) + H^{j-1}(K_1^F)$.

Schematy wykorzystujące opisaną technikę self-healing zapewniają *backward secrecy* oraz *forward secrecy*. Backward secrecy jest uzyskiwane przez zastosowanie łańcucha $(K_0^F, K_1^F, ..., K_m^F)$. Użytkownik U_i, który zostaje dodany do grupy multicastowej w sesji j, otrzymuje w ramach swojego klucza personalnego element K_j^F, a w związku z tym jest w stanie wyliczyć sekwencję $(K_j^F, ..., K_m^F)$, natomiast nie jest w stanie uzyskać żadnego z poprzednich elementów $(K_1^F, ..., K_{j-1}^F)$. Forward secrecy wynika z zastosowania selektywnej dystrybucji elementów łańcucha $(K_0^B, K_1^B, ..., K_m^B)$. W sesji j tylko użytkownicy należący do grupy G_j są w stanie odzyskać K_{m-j+1}^B i wyliczyć sekwencję $(K_{m-k+2}^B, ..., K_m^B)$. Ponieważ funkcja H jest jednokierunkowa, nie są oni w stanie uzyskać żadnego z elementów $(K_1^B, ..., K_{m-j}^B)$, które mogłyby zostać użyte do wyliczenia przyszłych kluczy sesyjnych.

Przedstawiona metoda jest bardzo wydajna, z punktu widzenia obciążenia komunikacyjnego, ponieważ w odróżnieniu od metody zastosowanej w schemacie bazowym nie wymaga ona retransmisji danych z poprzednich sesji. Ma również niewielkie wymagania pamięciowe – użytkownicy muszą dodatkowo przechowywać tylko pojedynczy element łańcucha *forward chain*. Niestety nie może być ona stosowana w schematach o nieograniczonej liczbie sesji, ponieważ wymaga wygenerowania w czasie inicjalizacji odpowiednio długich łańcuchów mieszających. Ponadto, koszt obliczeniowy wynikający z wyliczania funkcji skrótu jest zacznie większy niż koszt operacji na wielomianach.

Główną słabością mechanizmu *self-healing* opartego na łańcuchach mieszających jest brak odporności na *collusion attack*. Dwóch dowolnych użytkowników U_1 i U_2, takich że $U_1 \in R_l$ został wykluczony z grupy w sesji l, natomiast $U_2 \in J_r$ został dodany do grupy w sesji r, oraz $l < r$, jest w stanie współpracując ze sobą wyliczyć wartości kluczy sesyjnych K_l, ..., K_{r-1}, do których żaden z nich nie ma uprawnień. Wynika to z faktu, że po wykluczeniu z grupy U_1 jest wciąż w stanie wyliczać elementy łańcucha $(K_l^F, ..., K_m^F)$, natomiast U_2 po otrzymaniu K_{m-r+1}^B jest w stanie wyliczyć wszystkie poprzednie elementy $(K_{m-r+2}^B, ..., K_m^B)$. Pojawiło się wiele publikacji podejmujących próbę zapewnienia collusion resistance w schematach wykorzystujących łańcuchy mieszające, między innymi [18, 26, 9, 14], ale żaden z zaproponowanych mechanizmów nie rozwiązuje w pełni tego problemu. Co więcej, według nas zapewnienie *colussion reistance* nie jest możliwe bez utraty korzystnych właściwości mechanizmów opartych na łańcuchach mieszających.

4. Podsumowanie

W niniejszym artykule zaprezentowano ogólną ideę schematów SH-GKDS oraz kryteria analizy istniejących

rozwiązań. Szczegółowo opisano schemat bazowy, który stanowi punkt wyjścia do opisu i analizy najważniejszych mechanizmów stosowanych w schematach SH-GKDS.

Przeprowadzona analiza pokazuje, że projektowanie bezpiecznych schematów grupowej dystrybucji kluczy nie jest rzeczą łatwą i mimo obszernej bazy literaturowej nadal istnieje potrzeba opracowania bezpiecznego, a równocześnie wydajnego schematu SH-GKDS. Żaden z istniejących schematów nie jest wystarczająco dojrzały, aby mógł być użyty w praktycznych zastosowaniach. W większości istniejących rozwiązań, maksymalny czas działania systemu jest ograniczony i musi zostać precyzyjnie oszacowany już podczas inicjalizacji systemu. Istnieją dwa czynniki ograniczające maksymalny czas działania systemu SH-GKDS, są to: maksymalna liczba sesji oraz maksymalna liczba użytkowników, którzy mogą zostać wykluczeni z grupy w całym czasie działania systemu. W większości zastosowań określenie zawczasu wymaganej liczby sesji, którą system będzie musiał obsłużyć oraz całkowitej liczby użytkowników, którzy będą mogli zostać wykluczeni z grupy jest trudne, ponieważ dopuszczalne wartości tych dwóch parametrów zależą od dynamiki składu grupy, oraz od zachowania węzłów – członków grupy (zgodności ich zachowania z zasadami obowiązującymi grupę). Co więcej, nie wolno znacznie zawyżyć wartości tych parametrów, ponieważ mają one bezpośredni wpływ na koszt komunikacyjny procesu dystrybucji kluczy oraz na wymagania pamięciowe węzłów.

Zwiększenie czasu działania systemów SH-GKDS przy zachowaniu skalowalności oraz bezpieczeństwa systemu stanowi ważny i wciąż nierozwiązany problem.

Niniejsza praca została wykonana w ramach grantu nr 2281/B/T02/2008/35 finansowanego przez Ministerstwo Nauki i Szkolnictwa Wyższego.

Literatura

[1] Staddon J. *et al*.: *Self-Healing Key Distribution with Revocation*. IEEE Symposium on Security and Privacy, p. 241, May 2002

[2] Liu D., Ning P., Sun K.: *Efficient self-healing group key distribution with revocation capability*. [w:] Proceedings of the 10th ACM conference on Computer and communications security, New York, NY, USA, 2003, s. 231–240

[3] Blundo C., D'arco P., Santis A.D., Listo M.: *Design of Self-Healing Key Distribution Schemes*. Des. Codes Cryptography, vol. 32, s. 15–44, May 2004

[4] Hong D., Kang J.-S.: *An efficient key distribution scheme with self-healing property*. Communications Letters, IEEE, vol. 9, No. 8, s. 759–761, Aug. 2005

[5] Dutta R., Wu Y.D., Mukhopadhyay S.: *Constant Storage Self-Healing Key Distribution with Revocation in Wireless Sensor Network*. [w:] Communications, 2007. ICC '07. IEEE International Conference on, Glasgow, Scotlan, Jun. 2007, s. 1323–1328

[6] Dutta R., Mukhopadhyay S.: *Improved Self-Healing Key Distribution with Revocation in Wireless Sensor Network*. [w:] Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE, Kowloon, China, Mar. 2007, s. 2963–2968

[7] Daza V., Herranz J., Saez G.: *Flaws in some self-healing key distribution schemes with revocation*. Inf. Process. Lett., vol. 109, s. 523–526, May 2009

[8] Yuan T., Ma J., Zhong Y., Zhang S.: *Self-Healing Key Distribution with Revocation and Collusion Resistance for Wireless Sensor Networks*. [w:] Proceedings of the 2008 International Multi-symposiums on Computer and Computational Sciences, Washington, DC, USA, Oct. 2008, s. 83–90

[9] Dutta R., Mukhopadhyay S., Dowling T.: *Trade-off between collusion resistance and user life cycle in self-healing key distributions with t-revocation*. [w:] Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the, London, UK, Aug. 2009, s. 603–608

[10] Dutta R., Mukhopadhyay S., Collier M.: *Computationally secure self-healing key distribution with revocation in wireless ad hoc networks*. Ad Hoc Networks, vol. 8, No. 6, s. 597–613, 2010

[11] Han S., Tian B., He M., Chang E.: *Efficient threshold self-healing key distribution with sponsorization for infrastructureless wireless networks*. Wireless Communications, IEEE Transactions on, vol. 8, s. 1876–1887, Apr. 2009

[12] Zou X., Dai Y.-S.: *A Robust and Stateless Self--Healing Group Key Management Scheme*. [w:] Communication Technology, 2006. ICCT '06. International Conference on, Guilin, China, Nov. 2006, s. 1–4

[13] Tian B., Han S., Dillon T.S.: *An Efficient Self-Healing Key Distribution Scheme*. [w:] New Technologies, Mobility and Security, 2008. NTMS '08, Tangier, Morocco, Nov. 2008, s. 1–5

[14] Dutta R., Mukhopadhyay S., Dowling T.: *Enhanced Access Polynomial Based Self-healing Key Distribution*. [w:] Security in Emerging Wireless Communication and Networking Systems, O. Akan, *et al.*, Eds. Springer Berlin Heidelberg, 2010, vol. 42, s. 13–24

[15] Yuan T, Ma J., Zhong Y., Zhang S.: *Efficient Self--Healing Key Distribution with Limited Group Membership for Communication-Constrained Networks*. [w:] Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 1, Washington, DC, USA, 2008, s. 453–458

[16] Yuan T., Ma J., Zhong Y., Zhang S.: *Self-Healing Key Distribution with Limited Group Membership Property*. [w:] Proceedings of the 2008 First International Conference on Intelligent Networks and Intelligent Systems, Washington, DC, USA, Nov. 2008, s. 309–312

[17] Saez G.: *On Threshold Self-healing Key Distribution Schemes*. [w:] Cryptography and Coding, N. Smart, Ed. Springer Berlin / Heidelberg, 2005, vol. 3796, s. 340–354

[18] Tian B., Han S., Dillon T.S., Das S.: *A self-healing key distribution scheme based on vector space secret sharing and one way hash chains*. [w:] Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks, Washington, DC, USA, Jun. 2008, ps. 1–6

[19] Dutta R., Mukhopadhyay S., Das A., Emmanuel S.: *Generalized self-healing key distribution using vector space access structure*. [w:] Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet, Berlin, Heidelberg, 2008, s. 612–623

[20] Gu J., Xue Z.: *An Efficient Self-Healing Key Distribution with Resistance to the Collusion Attack for Wireless Sensor Networks*. [w:] Communications (ICC), 2010 IEEE International Conference on, Cape Town, South Afric, May 2010, s. 1–5

[21] Du X., Wang Y., Ge J., Wang Y.: *An ID-based broadcast encryption scheme for key distribution*. Broadcasting, IEEE Transactions on, vol. 51, No. 2, s. 264–266, Jun. 2005

[22] Tian B., Han S., Dillon T.S.: *A Self-Healing and Mutual-Healing Key Distribution Scheme Using Bilinear Pairings for Wireless Networks*. [w:] Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on, vol. 2, Shanghai, China, Dec. 2008, s. 208–215

[23] Han S., Tian B., Zhang Y., Hu J.: *An Efficient Self-Healing Key Distribution Scheme with Constant-Size Personal Keys for Wireless Sensor Networks*. [w:] Communications (ICC), 2010 IEEE International Conference on, Cape Town, South Africa, May 2010, s. 1–5

[24] Jiang Y., Lin C., Shi M., Shen X.: *Self-healing group key distribution with time-limited node revocation for wireless sensor networks*. Ad Hoc Networks, vol. 5, No. 1, s. 14–23, 2007, Security Issues in Sensor and Ad Hoc Networks

[25] Dutta R., Chang E.-C., Mukhopadhyay S.: *Efficient Self-healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Key Chains*. [w:] Applied Cryptography and Network Security, J. Katz and M. Yung, Eds. Springer Berlin / Heidelberg, 2007, vol. 4521, s. 385–400

[26] ChunLai H.Z., Zhang W.: *Anti-collusive Self-healing Key Distribution Scheme with Revocation Capability*. Information Technology Journal, vol. 8, s. 619–624, 2009

[27] Tian B., Han S., Parvin S., Hu J., Das S.: *Self-Healing Key Distribution Schemes for Wireless Networks: A Survey*. The Computer Journal, vol. 54, No. 4, s. 549–569, Mar. 2011